

TITLE OF THE INVENTION
INFORMATION PROCESSING SYSTEM, APPARATUS, AND METHOD,
AND STORAGE MEDIUM

5 FIELD OF THE INVENTION

The present invention relates to an information processing system, apparatus, and method for providing a service on a network, and a storage medium.

10 BACKGROUND OF THE INVENTION

Systems for setting a service provider which provides a service on a network, searching for or exploring the provided service on the network, and allowing the user to use the service provided by the
15 service provider have recently been developed. These systems provide services by using Jini, UPnP (Universal Plug and Play), SLP (Service Location Protocol), or the like (Jini is a trademark of Sun Microsystems).

As a typical example, a service
20 registration/search/use method using Jini will be described.

Figs. 12, 13, 14, and 15 are views showing a method of providing a service by Jini. In Figs. 12 to 15, reference numeral 2010 denotes a network which is
25 represented by the Internet and controlled by the network protocol TCP/IP; 2020, a service provider which provides a service to another node connected to the

network; 2030, a Lookup server which provides a Lookup
service for registering and searching for a service;
2040, a client which uses a service; 2050, a proxy
object executed as a proxy of the service provider on
5 the client; and 2060, a service attribute representing
the function of a service used to search for a service.

(1) Search of Lookup Service

As shown in Fig. 12, a server called the Lookup
server 2030 in which services are registered operates on
10 the network. This server provides service
registration/search as a service (to be referred to as a
Lookup service) to the service provider 2020 and the
client 2040 serving as a service user.

A Discovery protocol as a network protocol used by
15 the service provider 2020 and client 2040 in order to
search for the Lookup server 2030 on the network is
defined. The Discovery protocol is defined as a
multicast protocol of TCP/IP, and broadcasted to a
plurality of nodes on the network.

20 The Lookup server 2030 which has received a
Discovery protocol message sends back a response message
to the node which sent the message. The response message
includes a class object having a built-in Java Interface
(ServiceRegistrar) designed to function as the Lookup
25 service for registering and searching for a service
(Java is a trademark of Sun Microsystems).

The service provider 2020 and client 2040 which

have received the response message extract the
ServiceRegistrar built-in object from the response
message, and use it as a proxy object of the Lookup
service which provides service registration/search
5 services.

(2) Service Registration

Fig. 13 shows a service registration method. The
service provider 2020 uses a Lookup service proxy object
to register a service provided by the service provider
10 2020 itself.

The Lookup service proxy object has a built-in
register method as a service registration means. The
register method receives as arguments the proxy object
2050 executed as a proxy object on the client side in
15 order to provide a service to the client, and the
service attributes 2060 which represent the functions of
the service to be provided or the like. Then, the proxy
object 2050 and service attributes 2060 are transmitted
to the Lookup server 2030. The Lookup server 2030 holds
20 and manages the proxy object 2050 and service attributes
2060 so as to provide them in accordance with a request
from the client.

Note that a protocol for transmitting the proxy
object 2050 and service attributes 2060 by the register
25 method is not defined in Jini, and depends on the
built-in ServiceRegistrar.

(3) Service Search

Fig. 14 shows a service search method. The client 2040 uses the Lookup service proxy object to search for and acquire a service to be used.

The Lookup service proxy object has a built-in
5 lookup method as a service search/acquisition means. The
lookup method receives as arguments a Java interface
which defines a program interface for identifying the
proxy object 2050 executed as a proxy object on the
client side in order to provide a service and providing
10 the function of the proxy object 2050, and a template
object in which the attributes of a service to be used
is built-in as a search condition. This template is
transmitted to the Lookup server 2030. The Lookup server
2030 searches held/managed service proxy objects 2050
15 and service attributes 2060 for a proxy object 2050
which meets the transmitted template condition. The
searched proxy object 2050 is sent back as the return
value of the lookup method to the client 2040.

Note that a protocol for transmitting the template
20 and proxy object 2050 by the lookup method is not
defined in Jini, and depends on the built-in
ServiceRegistrar.

(4) Service Use

Fig. 15 shows a service use method.

25 The client 2040 uses the proxy object 2050 to use
a service. Jini defines a Java interface as a
program/interface for using a service. The proxy object

2050 has the built-in Java interface. The client can receive a service by using the method provided by the interface.

How to build the defined interface into the proxy object 2050 depends on the service provider, and the Java interface built-in method does not influence the programming of the client. A service can be used regardless of the Lookup server 2030.

A problem to be solved by the present invention will be explained with reference to the example of Figs. 16 and 17. In Figs. 16 and 17, reference numeral 3010 denotes a client which receives a service; 3020, an A service provider which provides service A; 3030, an accounting service provider which provides an accounting service for paying fees for receiving a service; 3040, an A service proxy object serving as a proxy of the A service provider on the client 3010 for providing service A; and 3050, an accounting service proxy object 3050 serving as a proxy of the accounting service provider 3030 on the client 3010 in order to provide an accounting service.

In a normal mechanism, as shown in Fig. 16, the client 3010 acquires proxy objects for the A service and accounting service, i.e., the A service proxy object 3040 and accounting service proxy object 3050, and in exchanging information between services, must relay the information exchange, which increases the load on the

client.

To solve this, according to the present invention as shown in Fig. 17, the accounting service proxy object 3050 is transferred to the A service provider 3020,

5 personal information of the client 3010 is set via the A service provider 3020, and the A service provider 3020 directly uses the accounting service provided by the accounting service provider 3030.

In this configuration, the accounting service is
10 received via the A service provider, which may cause leakage of client's personal information. Further, since the object which receives a service is different from the A service provider, the accounting service provider does not recognize that the A service provider 3020 uses
15 the accounting service provider 3030 for the client 3010, but recognizes that the A service provider 3020 itself newly uses the accounting service provider 3030.

SUMMARY OF THE INVENTION

20 The present invention has been made to overcome the conventional drawbacks, and has as its object to reduce the processing burden on the user in using a second service of a different service provider together with the use of a first service.

25 It is another object of the present invention to improve the security when the second service requests personal information in using the second service

together with the use of the first service.

5 The present invention provides an information processing method executed by the information processing system and apparatus. In addition, the present invention provides a storage medium which stores a control program for implementing the information processing system or apparatus by a computer.

10 Other features and advantages of the present invention will be apparent from the following description taken in conjunction with the accompanying drawings, in which like reference characters designate the same or similar parts throughout the figures thereof.

15

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram showing the arrangement of an information service system according to the first embodiment;

20 Fig. 2 is a view showing the structure of an accounting service proxy object according to the first embodiment;

Fig. 3 is a flow chart for explaining the processing sequence of the information service system according to the first embodiment;

25

Fig. 4 is a view for explaining a service providing form according to the first embodiment;

providing form in the use of another service; and

Fig. 17 is a view showing another general service providing form in the use of another service.

The accompanying drawings, which are incorporated
5 in and constitute a part of the specification,
illustrate embodiments of the invention and, together
with the description, serve to explain the principles of
the invention.

10 DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Preferred embodiments of the present invention
will now be described in detail in accordance with the
accompanying drawings.

(First Embodiment)

15 The first embodiment will be described by
exemplifying a case using the Java language. However, it
will be apparent to those skilled in the art from the
following description that the present invention does
not depend on the Java language and can be implemented
20 by using another programming language. The following
embodiment will be explained using a combination of a
given service and accounting service. However, the
accounting service need not always be included, and the
present invention can be applied to a combination of any
25 services.

Fig. 1 is a block diagram showing the arrangement
of an information service system according to the first

embodiment. In Fig. 1, reference numeral 101 denotes a network which connects a plurality of information processing apparatuses and is the Internet in the first embodiment; 102, a client which uses each service; 103, a service provider which provides various services; and 104, an accounting service provider which provides an accounting service. The service provider 103 and accounting service provider 104 provide services by using Jini. Hence, the client 102 receives the provided services by using Jini.

Reference numeral 105 denotes a Lookup server which registers the services of the service provider 103, accounting service provider 104, and the like, and searches for and provides registered services in accordance with a request from the client.

The client 102, service provider 103, accounting service provider 104, and Lookup server 105 are respectively constituted by computers. A typical arrangement is illustrated for the client 102. A CPU 111 executes various control operations in accordance with a control program stored in a memory. A ROM 112, RAM 113, and external memory 114 provide memories for providing the control program and various work areas of the CPU 111. An input device 115 is made up of a keyboard, pointing device, and the like, and allows the operator to input various inputs. Reference numeral 116 denotes a display which displays various displays under the

control of the CPU 111; and 117, a network interface which connects the network 101 to the computer.

Although the typical arrangement is illustrated for the client 102, the computers of the service provider 103, accounting service provider 104, and Lookup server 105 also have similar arrangements.

Fig. 2 is a view showing the structure of an accounting service proxy object according to the first embodiment. In Fig. 2, reference numeral 202 denotes an accounting service proxy object; and 203, client information for identifying a client which uses an accounting service.

The first embodiment releases the client from the burden of relaying information exchange between the service of an A service proxy object as shown in Fig. 16 and the service of the accounting service proxy object, and prevents leakage of personal information in the system shown in Fig. 17. To prevent leakage of personal information, the client 102 sets client information to the accounting service provider, and the A service provider 103 uses the accounting service while taking over the state in which the client information is set.

For this purpose, according to the first embodiment, the client 102 acquires the accounting service proxy object 202 and sets the client information 203 in the accounting service proxy object 202, and the accounting service proxy object 202 holds the client

information 203. The accounting service proxy object 202
in which the client information 203 is set is provided
to the A service provider 103, thus handing over the
service. The operation of the first embodiment will be
5 explained in detail.

A case wherein the accounting service of the
accounting service provider 104 is used in using the A
service of the A service provider 103 will be described.

Fig. 3 is a flow chart for explaining the
10 processing sequence of the information service system
according to the first embodiment. Figs. 4 and 5 are
views for explaining a service providing form according
to the first embodiment. Fig. 6 is a view showing an
example of the accounting service proxy object according
15 to the first embodiment. In this example, the service
provided by the A service provider 103 assumes to
require the accounting service provided by the
accounting service provider 104.

In step S301, the client 102 acquires an A service
20 proxy object 201 from the Lookup server 105 by the
method described with reference to Figs. 13 and 14 (see
Fig. 4). If the A service proxy object 201 is executed
to find the necessity for the accounting service of the
accounting service provider 104, the client 102 acquires
25 the accounting service proxy object 202 from the Lookup
server 105 in step S302, similar to step S301 (see
Fig. 4). In the first embodiment, the client 102

acquires the A service proxy object 201 and accounting service proxy object 202 from the single Lookup server 105, but the present invention is not limited to this. That is, the Lookup server from which the A service proxy object 201 is acquired in step S301 need not be the same as the Lookup server from which the accounting service proxy object 202 is acquired in step S302.

In step S303, the client program sets the client information 203 by the setUserInformation method (see Fig. 6) of the accounting service proxy object 202 (see Fig. 4). In this setting, the accounting service provider 104 is not notified of the client information 203, and the client information 203 is held in the accounting service proxy object 202. Setting of the client information 203 will be explained in detail. For example, the contents of the client information 203 are a bank name, account number, and password used for payment. The pieces of information are set in the UserInformation object in the setUserInformation method of Fig. 6, and transferred to the accounting service proxy object 202. The accounting service proxy object 202 encrypts the UserInformation object (client information 203) by using a public key or the like, and holds it as the internal information of the accounting service proxy object 202.

The held client information 203 cannot be directly referred from another method or object, and the user

variable cannot be accessed by an object other than the accounting service proxy object. Thus, the A service provider cannot directly access or refer to the client information 203.

5 In step S304, the client 102 transfers the accounting service proxy object 202 to the A service provider 103 via the A service proxy object 201. In other words, the client 102 hands over the accounting service proxy object 202 to the A service provider 103 so as to allow the A service provider 103 to use the accounting service of the accounting service provider 104 (see Fig. 5). In handing over the accounting service proxy object 202, the accounting service proxy object 202 is serialized by the RMI (Remote Method Invocation) and Serialize mechanisms of the Java language execution environment and sent as network data (serial data) to the A service provider 103 where the accounting service proxy object to which the client information is set as an inside information, is reconstructed based on the serialized data. This inhibits the A service provider from directly accessing the client information 203.

 In step S305, the A service provider 103 uses the accounting service of the accounting service provider 104 via the accounting service proxy object 202 handed over from the client 102. As represented by the pay method in Fig. 6, the A service provider uses the accounting service by using the client information 203

set in the accounting service proxy object 202 by the client 102 in step S303 as if the client 102 used the accounting service.

In the pay method, the encrypted client
5 information 203 held by the accounting service proxy object 202 is transferred to the accounting service provider 104. The accounting service provider 104 decrypts the encrypted accounting information and the like, and performs debit processing from the bank
10 account on the basis of the pieces of information.

In the first embodiment, the accounting service proxy object 202 encrypts the client information 203 in step S303. Alternatively, when the client information 203 is set in the UserInformation object, it may be
15 temporarily transferred to and encrypted by the accounting service provider 104, and then held in the accounting service proxy object 202.

The first embodiment has exemplified the accounting service, but another service can also be
20 handed over by the same method. For example, personal authentication can be received from an authentication service and handed over to another service to guarantee that personal authentication is granted by the service.

As described above, according to the first
25 embodiment, the client need not relay the first and second services in using the second service (accounting service) along with the use of the first service (A

service).

According to the first embodiment, the proxy object of the second service holds necessary personal information, and the use of the second service is handed
5 over to the provider of the first service. The security is improved without laying open any personal information to the provider of the first service.

(Second Embodiment)

The second embodiment will be described. In the
10 first embodiment, the client 102 sets the client information 203 in the accounting service proxy object 202, and the A service provider 103 uses the accounting service by exploiting the client information 203, thereby improving the security of client information. In
15 the second embodiment, the security is improved by using a session ID established between the client and the accounting service provider, in addition to client information.

The arrangement of an information service system
20 according to the second embodiment is the same as that according to the first embodiment (Fig. 1). The second embodiment will also exemplify a case using the Java language. However, the present invention does not depend on the Java language and can be implemented by using
25 another programming language.

A case wherein a program executed by a client 102 uses the service of an A service provider 103 will be

exemplified with reference to Figs. 7 to 11. Similar to the first embodiment, the service of the A service provider 103 uses that of an accounting service provider 104.

5 Fig. 7 is a flow chart for explaining the processing sequence of the information service system according to the second embodiment. Figs. 8 and 9 are views for explaining a service providing form according to the second embodiment. Fig. 10 is a table for
10 managing a pair of a session ID and client information used within the accounting service provider. Fig. 11 is a view showing an example of an accounting service proxy object according to the second embodiment.

 In step S701, the client 102 acquires an A service
15 proxy object 201 from a Lookup server 105 by the method described with reference to Figs. 13 and 14 (see Fig. 8). In step S702, similar to step S701, the client 102 acquires an accounting service proxy object 202 from the Lookup server 105 (see Fig. 8). In the second embodiment,
20 the client 102 acquires the A service proxy object 201 and accounting service proxy object 202 from the single Lookup server 105, but the present invention is not limited to this. That is, the Lookup server from which the A service proxy object 201 is acquired in step S701
25 need not be the same as the Lookup server from which the accounting service proxy object 202 is acquired in step S702.

In step S703, the client program sets client information 203 by the setUserInformation method (see Fig. 11) of the accounting service proxy object 202 (see Fig. 8). In this setting, the accounting service provider 104 is notified of the client information 203 via the accounting service proxy object 202. The accounting service provider 104 which has been notified of the client information 203 interprets that a series of accounting service sessions newly start, generates a new session ID 801, and sends it back to the client 102 via the accounting service proxy object 202.

Note that the accounting service provider 104 registers in the table the notified client information 203 and the session ID 801 assigned to the client information 203, and uses them for authentication in providing a service (see Fig. 10).

In step S704, the client 102 hands over the accounting service proxy object 202 and the session ID 801 acquired from the accounting service to the A service provider 103 so as to use the accounting service proxy object 202 via the A service proxy object 201 (see Fig. 9). In handing over the accounting service proxy object 202 and session ID 801, the accounting service proxy object 202 and session ID 801 are serialized by the RMI and Serialize mechanisms of the Java language execution environment and sent as network data to the A service provider where the accounting service proxy

object 202 to which the client information is set as an inside information, and the session ID 801 are reconstructed based on the serialized data.

In step S705, the A service provider 103 uses the accounting service via the accounting service proxy object 202 handed over from the client 102. As represented by the pay method in Fig. 11, the A service provider 103 uses the accounting service of the accounting service provider 104 by using the session ID 801 acquired from the accounting service provider 104 in step S704. Authentication is executed within the accounting service provider 104 by checking the session ID and client information with reference to the table shown in Fig. 10, and the A service provider 103 executes the accounting service as if the client 102 used the accounting service. For example, when the contents of the client information 203 are a bank name, account number, and password used for payment, the pieces of information are stored in client information 1002 of Fig. 10. In the pay method, since the session ID has been transferred, the session ID and client information 203 are sent to the accounting service provider. The accounting service provider 104 refers to the table of Fig. 10 by using the received session ID as a key, obtains client information, and collates the received client information 203 with the client information. If collation is "OK", debit processing from

the bank account is performed in accordance with the client information 203.

The second embodiment has exemplified the accounting service, but another service can also be
5 handed over by the same method.

As described above, according to the second embodiment, the use of a session ID established between the client and the accounting service provider in addition to client information can improve the security
10 more than the arrangement of the first embodiment.

According to each of the above embodiments, a service necessary for the client that is searched for by the client can be provided as if the client received another service.

15 The object of the present invention is also achieved when a storage medium which stores software program codes for realizing the functions of the above-described embodiments is supplied to a system or apparatus, and the computer (or the CPU or MPU) of the
20 system or apparatus reads out and executes the program codes stored in the storage medium.

In this case, the program codes read out from the storage medium realize the functions of the above-described embodiments, and the storage medium which
25 stores the program codes constitutes the present invention.

The storage medium for supplying the program codes

includes a floppy disk, hard disk, optical disk, magnetooptical disk, CD-ROM, CD-R, magnetic tape, nonvolatile memory card, and ROM.

5 The functions of the above-described embodiments are realized not only when the computer executes the readout program codes, but also when the OS (Operating System) running on the computer performs part or all of actual processing on the basis of the instructions of the program codes.

10 The functions of the above-described embodiments are also realized when the program codes read out from the storage medium are written in the memory of a function expansion board inserted into the computer or the memory of a function expansion unit connected to the
15 computer, and the CPU of the function expansion board or function expansion unit performs part or all of actual processing on the basis of the instructions of the program codes.

20 As has been described above, the present invention can reduce the processing burden on the user in using a second service of a different service provider together with the use of a first service.

Also, the present invention can improve the security when the second service requests personal
25 information in using the second service together with the use of the first service.

As many apparently widely different embodiments of

the present invention can be made without departing from the spirit and scope thereof, it is to be understood that the invention is not limited to the specific embodiments thereof except as defined in the claims.

09900036.070901